specific functionalities, functional tasks, or task groupings that are in some cases arbitrarily assigned to the specific modules for explanatory purposes. It will be appreciated by the person having ordinary skill in the art that an RBAC system according to the present invention may be arranged in a variety of ways, or that functional tasks may be grouped according to other nomenclature or architecture than is used herein without doing violence to the spirit of the present invention.

[0029] Referring to **FIG. 1**, an RBAC model **11** known in the art and operating by user-role assignment, includes: users, or subjects, **13**, roles **15**, and access permissions **17** assigned to the roles **15**. Parts of **FIG. 1** not necessary to an explanation of the present invention will not be elaborated on but are assumed to be understood by a person having ordinary skill in the art. When a user **13** initiates a request for object access at the session controller **19**, the user **13** is verified as having a valid role **15**. The permissions **17** allow role access to the objects **21**, such as medical records, and determine which operations **23** the role **15** may perform on the objects **21**.

[0030] Referring to **FIG. 2**, the present invention presents a role-based access control system **24** for a controlled computer system having refined and dynamic permission constraints **25** which are tested against facts/data, i.e., content **27**, or contexts **29** derived from the content, achieved through the use of data extraction, e.g., known information retrieval, data mining, and natural language processing techniques, represented by the external database **31** and the internal database data extraction represented by dotted lines **33**, from each of the object **21**, role **15** and user **13** domains indicating data extraction. It will be noted that data extraction is not limited herein to metadata searching but includes the ability to obtain actual text or other content from within the selected data constructs. The exemplary embodiment **24** of the present invention illustrated in **FIG. 2** also includes users, or subjects, **13**, roles **15**, and access permissions **17** assigned to the roles **15**. After the user **13** is verified as having a valid role **15** by the session controller **19**, the user **13** initiates a request **26** for object access. The extracted data or content **27** may be gathered and compared to verify context **29** as set forth in the constraints **25**. Content **27** may be gathered and contexts **29** verified for each and every information category (subject, object, environment) individually, or contexts verified between categories, such as application context APP formed between object content and subject content, or system context SYS formed between subject content and environmental content. Before access **37** to objects **21** is granted, each constraint **25** on the role permission **35** must be verified to limit the retrieval of data, or other operations **23** on the objects **21**, to those intended by a system administrator (not shown). Constraints **25** on the role's permission **35** written about/against full content **27** and context **29** may then be tested and compared to each and every of the subject, object, or environment information. If the content and the context of the constraints are validated, access **37** is granted allowing the user (subject) **13** to receive permissible portions of the objects **21** and operate **23** upon them, such as view/copy/modify; according to the constraints **25** imposed on the role permission **35**.

[0031] The specific tools, functionalities, or applications necessary to accomplish the present invention are considered to be within the skill of the art. For example, possible languages to specify constraints may include, for example, SQL, Relational Algebra, or Prepositional Logic or similar functionalities now known or later developed. Possible data extraction techniques may include approaches that rely upon, for example, part of speech tagging, conventional term extraction, term co-occurrence, inference networks, language models, or similar functionalities now known or later developed. Possible search mechanisms for locating content or context may include, for example, crawlers, mediators, text search engines, database management systems search approaches as used for relational, hierarchical, or other logical database models, geospatial database search approaches, or reconciled structured repository (both logical and physical) search routines, or similar constructs or functionalities now known or later developed.

PERMISSION CONSTRAINT EXAMPLES

Example 1

[0032] Head nurses can view all their department doctors' patients' medical records, except the medical records of the immediate family of said head nurses' colleagues within the same department.

[0033] Such a determination of colleagues may require extensive user identity knowledge besides that available from the user profile provided at log-in to the session, i.e., prior to the access request. The determination of immediate family may even require retrieval of data external to the controlled computer system. A parenthetical category review of Example 1 shows: head nurses (a role, or subject information) can view (operation) all their department doctors' patients' medical records (ownership or object information and relationship context of doctor and nurse), except (constraint on access) the medical records (objects) of their colleague's immediate family (possible environment or subject information or both, and including content and context) in the same department.

[0034] Therefore, when:

[0035] User: U

[0036] Patient-Record: O

[0037] Roles: R={Patient, Nurse, HNurse, Doctor}

[0038] Operations: OP={view, append, copy}

[0039] Application Context:

[0040] Relationship:

[0041] ar=Affiliation Relationship

[0042] doctor=Doctor-Patient Relationship

[0043] fr=Immediate Family Relationship;

in a formal specification the role-Permission Assignment with Context Constraints may be written:

[0044] PA(HNurse, O, view) [[ar(doctor(owner(O))== ar(usr (HNurse)) && ar(fr(owner(O))) !=ar(usr(HNurse))]];

where:

[0045] [[]] represents the context constraints;

[0046] == is equal;